

Aprendizagem Automática Métodos de Agrupamento

G. Marques

Clustering

Os métodos de agrupamento, ou *clustering*, são técnicas de aprendizagem **não supervisionada** onde se procura encontrar representações mais compactas dos dados sem mais nenhuma informação sobre os mesmos. O objetivo é dividir os dados (vetores d -dimensionais) em grupos, de modo a que elementos do mesmo grupo sejam mais semelhantes entre si do que com membros de outros grupos. Isto implica definir uma função de semelhança (e.g. inverso da distância) e maximizar este valor mudando a atribuição dos grupos a cada ponto.

Estes métodos são extremamente úteis na exploração e representação de estruturas presentes nos dados. O cenário ideal é os dados estarem agrupados em nuvens multi-dimensionais de pontos, e nestas condições é provável que as observações pertencentes a um dado cluster tenham características comuns entre elas. Desta forma pode-se categorizar os dados e descobrir as “classes” que neles estão subjacentes, obtendo assim uma representação de alto nível dos dados. Esta representação mais compacta dos dados também é útil na compressão com perdas (quantificação vetorial) e na deteção de *outliers*.

De seguida, aborda-se o algoritmo k -médias e métodos de agrupamento hierárquico, os dendrogramas.

Clustering: k -Médias

O algoritmo k -médias é porventura um dos mais simples em aprendizagem automática. Esta técnica representa os dados através de k centroides em que cada ponto pertence ao centroide mais próximo. A localização dos centroides é feita iterativamente em dois passos, um de atribuição e outro de atualização.

Enquadramento

- Conjunto de dados $\mathcal{X} = \{\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[N]\}$
 $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \rightarrow$ vetor a d dimensões
- Objetivo: dividir dados em k clusters
 k definido ante mão
- Escolher uma métrica de distâncias entre dois vetores \mathbf{x} e \mathbf{c}
 - Euclidiana: $\text{dist}_{\ell_2}(\mathbf{x}, \mathbf{c}) = \|\mathbf{x} - \mathbf{c}\| = \sqrt{\sum_{i=1}^d (x_i - c_i)^2}$
 - Manhattan: $\text{dist}_{\ell_1}(\mathbf{x}, \mathbf{c}) = |\mathbf{x} - \mathbf{c}| = \sum_{i=1}^d |x_i - c_i|$
 - Cosseno: $\text{dist}_{\cos}(\mathbf{x}, \mathbf{c}) = 1 - \frac{\mathbf{x}^T \mathbf{c}}{\|\mathbf{x}\| \|\mathbf{c}\|} = 1 - \cos(\theta)$ θ : ângulo entre os vetores \mathbf{x} e \mathbf{c}
 - ...

Clustering: k -Médias

O algoritmo k -médias é porventura um dos mais simples em aprendizagem automática. Esta técnica representa os dados através de k centroides em que cada ponto pertence ao centroide mais próximo. A localização dos centroides é feita iterativamente em dois passos, um de atribuição e outro de atualização.

Pseudo-Código

- 1 Inicializar k centroides $\mathbf{c}[1], \dots, \mathbf{c}[K]$
- 2 ATRIBUIÇÃO: Calcular funções de pertença $r_1[n], \dots, r_k[n]$:

$$r_j[n] = \begin{cases} 1 & \text{se } j = \operatorname{argmin}_i \{\operatorname{dist}(\mathbf{c}[i], \mathbf{x}[n])\} \\ 0 & \text{caso contrário} \end{cases}$$

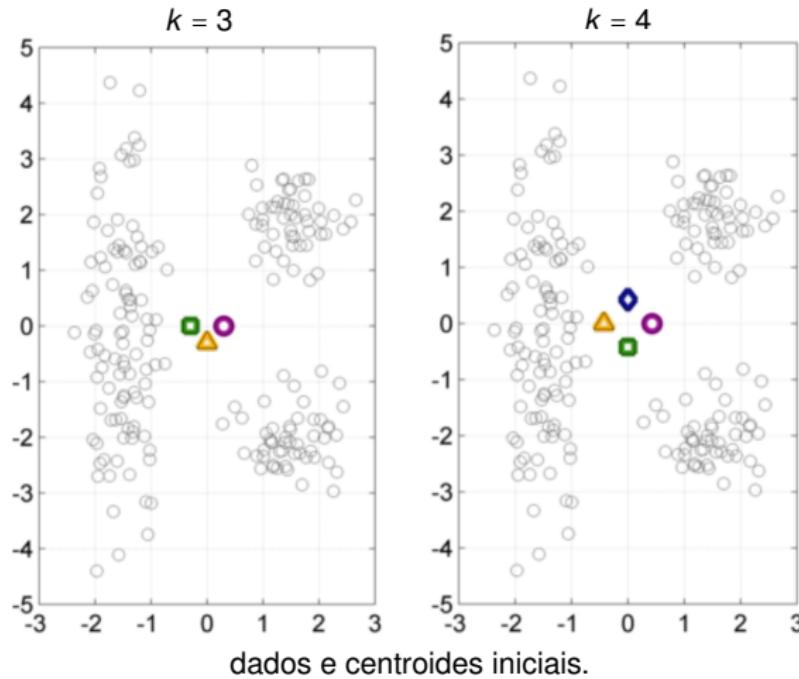
- 3 ATUALIZAÇÃO: Recalcular centroides.

$$\mathbf{c}_j = \frac{1}{R_j} \sum_{n=1}^N r_j[n] \mathbf{x}[n] \quad \text{com} \quad R_j = \sum_{n=1}^N r_j[n] = N_j$$

- 4 Repetir pontos 2 e 3.

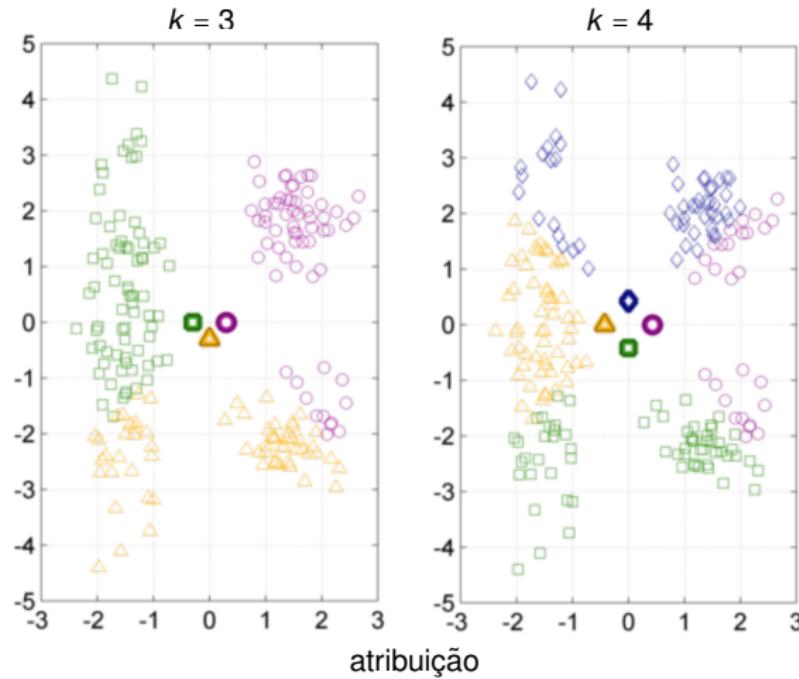
Clustering: k -Médias

Exemplo de dados sintéticos:



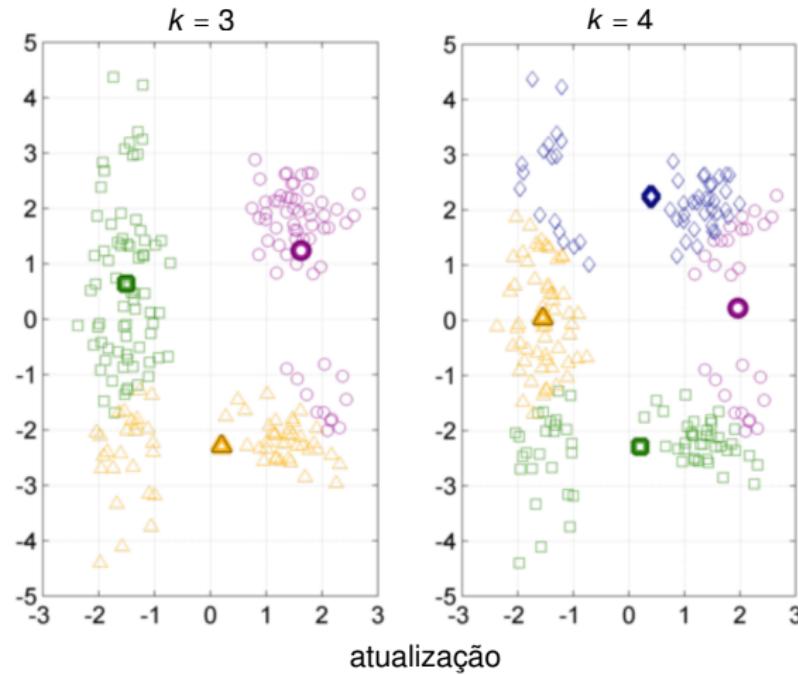
Clustering: k -Médias

Exemplo de dados sintéticos:



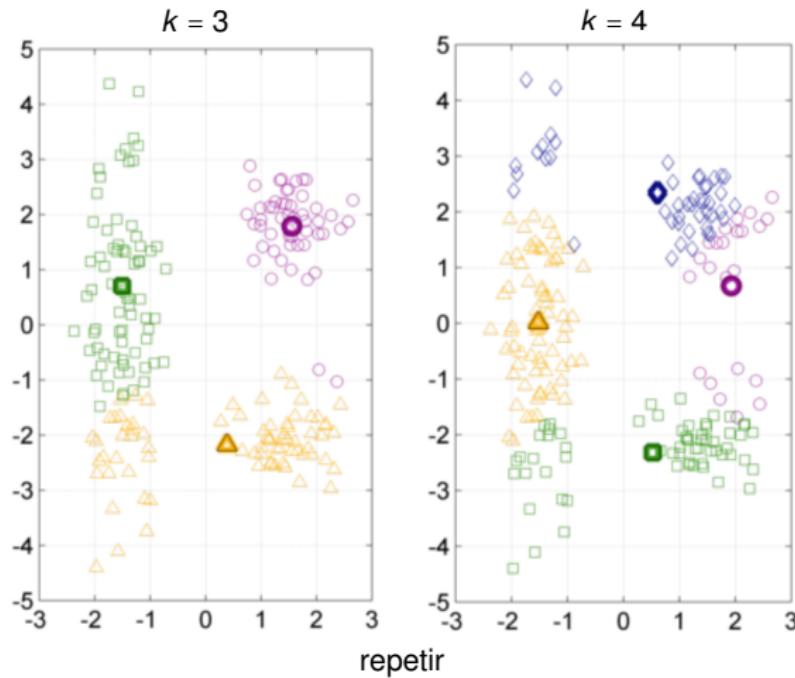
Clustering: k -Médias

Exemplo de dados sintéticos:



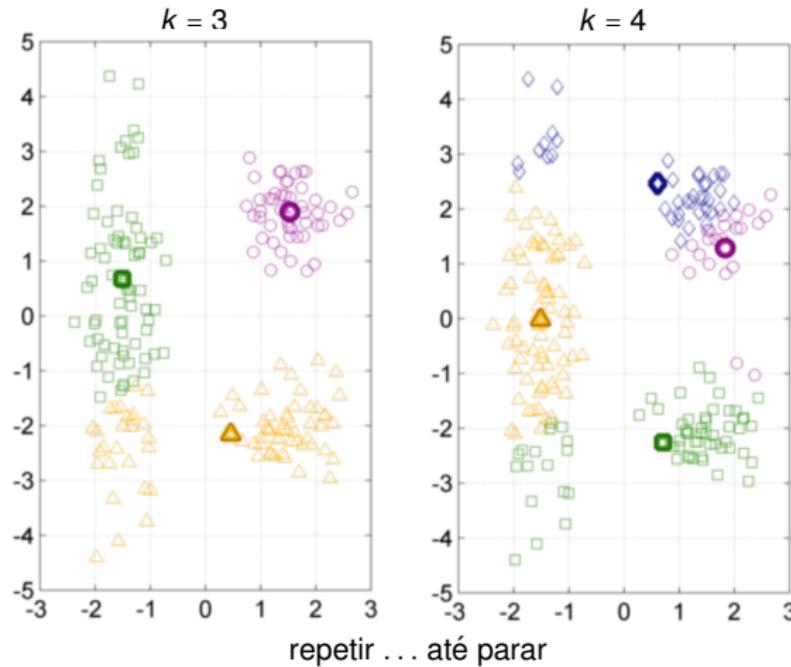
Clustering: k -Médias

Exemplo de dados sintéticos:



Clustering: k -Médias

Exemplo de dados sintéticos:

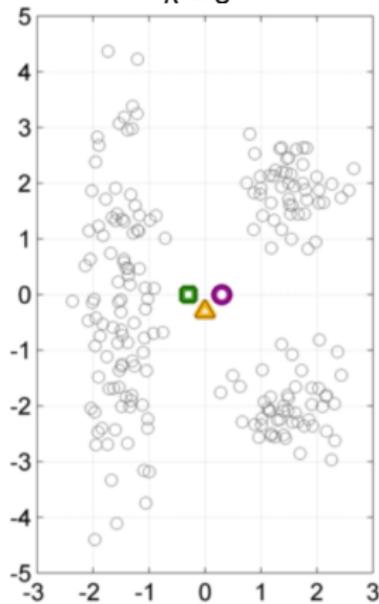


Clustering: k -Médias

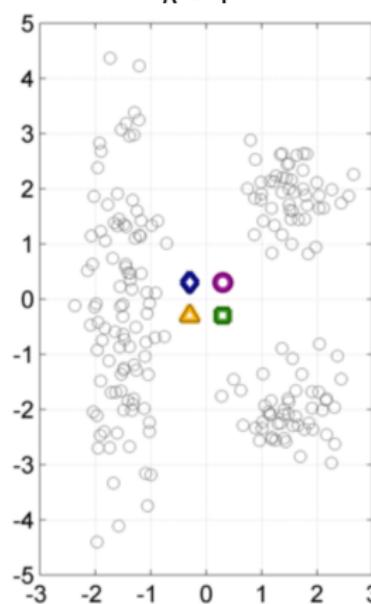
Exemplo de dados sintéticos:

Diferentes inicializações obtêm resultados diferentes!

$k = 3$



$k = 4$

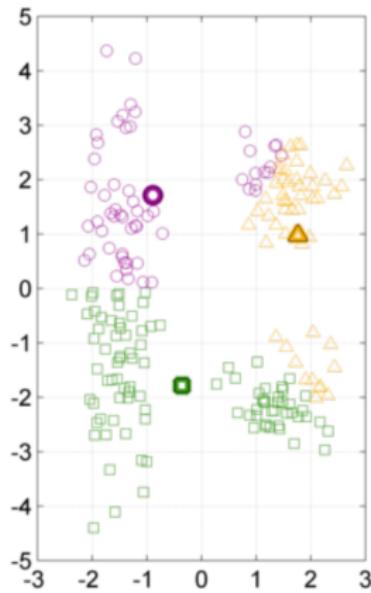


Clustering: k -Médias

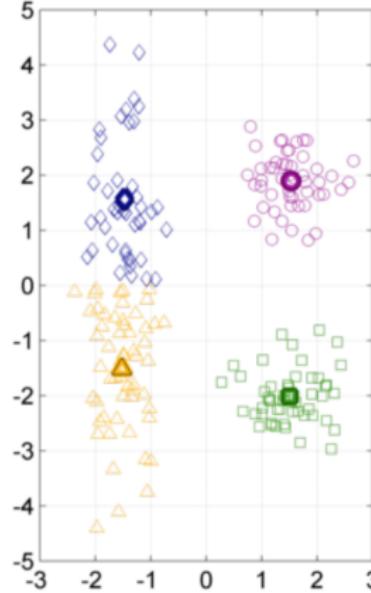
Exemplo de dados sintéticos:

Diferentes inicializações obtêm resultados diferentes!

$k = 3$



$k = 4$



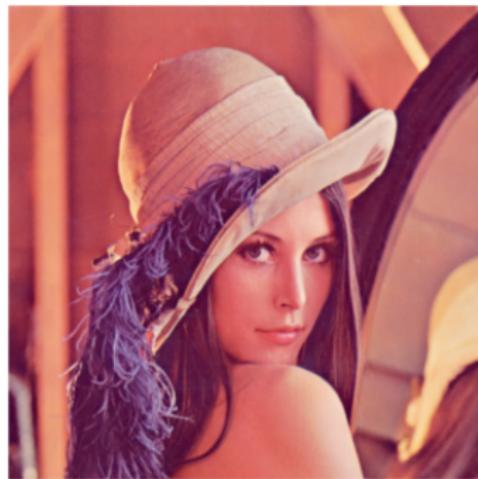
Clustering: k -Médias

Exemplo de quantificação de imagem:

- 3 planos com 8 bits por plano
- Cada pixel codificado com 24 bits
- Cada pixel é um ponto a 3 dimensões
- Converter imagem $L \times C \times 3$ em matriz $\mathbf{X} 3 \times (L \times C)$

Obter pontos 3D:

```
I=plt.imread('lena.tif')
lc=I.shape[0]*I.shape[1]
xr=I[:, :, 0]
xg=I[:, :, 1]
xb=I[:, :, 2]
X=np.zeros(3,lc)
X[0, :]=np.reshape(xr, (1, lc))
X[1, :]=np.reshape(xg, (1, lc))
X[2, :]=np.reshape(xb, (1, lc))
```

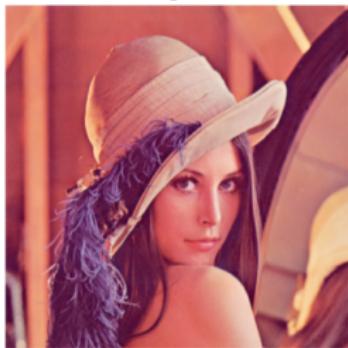


Clustering: k -Médias

Exemplo de quantificação de imagem:

- Representar imagem com diferentes níveis de quantificação:

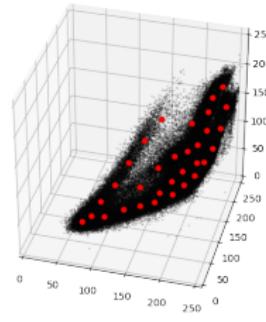
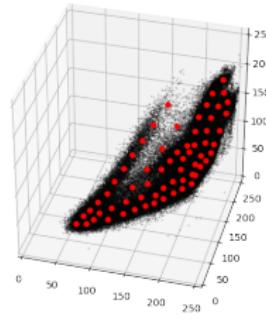
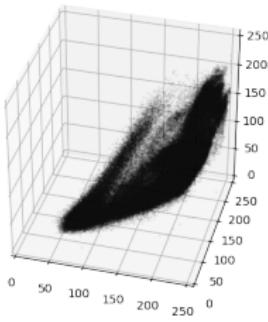
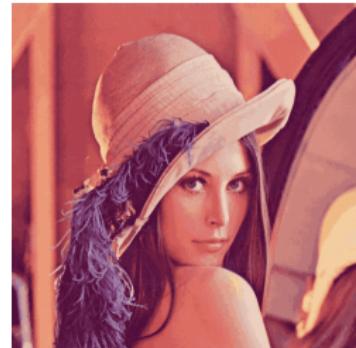
Original



$k=64$



$k=32$

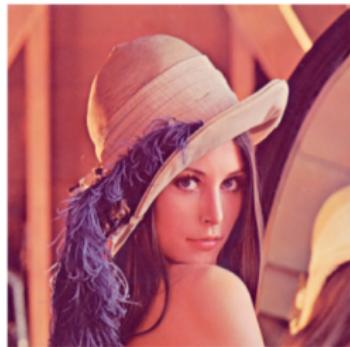


Clustering: k -Médias

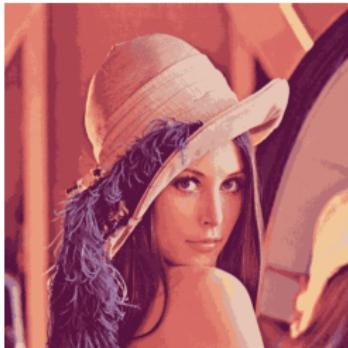
Exemplo de quantificação de imagem:

- Representar imagem com diferentes níveis de quantificação:

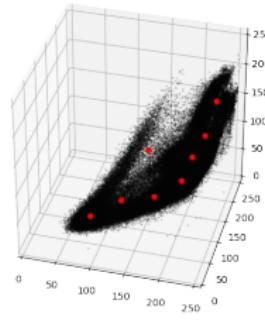
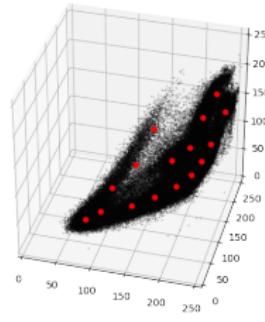
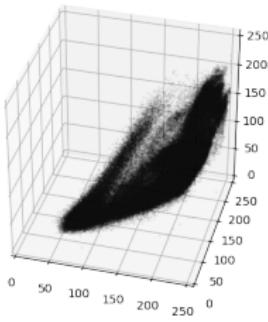
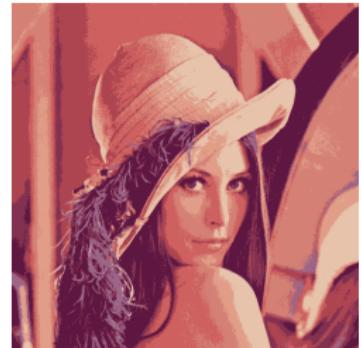
Original



$k = 16$



$k = 8$

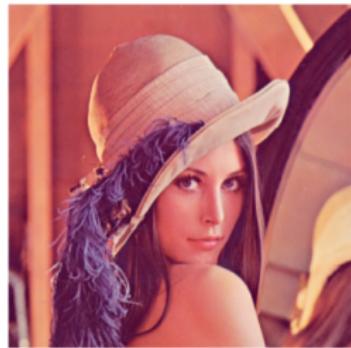


Clustering: k -Médias

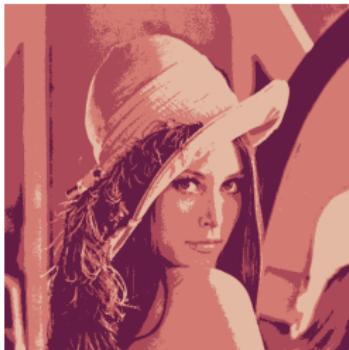
Exemplo de quantificação de imagem:

- Representar imagem com diferentes níveis de quantificação:

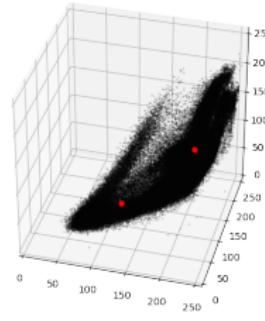
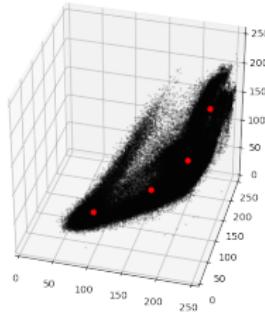
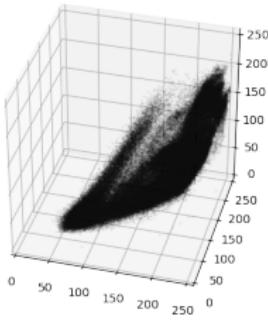
Original



$k=4$



$k=2$



Clustering: *k*-Médias sklearn

O algoritmo disponibilizado pelo `sklearn` é uma versão melhorada do *k*-médias, nomeadamente em termos inicialização dos centroides.

- Os próximos comandos, carregam e inicializam um classificador *k*-médias com $k=8$ centroides.

```
from sklearn.cluster import KMeans  
kmeans=KMeans(n_clusters=8, init='k-means++', n_init=5, \  
               max_iter=500, tol=0.0001, verbose=1, \  
               random_state=42)
```

- Aqui estão alguns parâmetros a ter em conta:
 - `n_clusters`: número de centroides (`default=8`).
 - `init`: método de inicialização dos centroides.
 - `'k-means++'`: algoritmo melhorado de seleção (`default`).
 - `'random'`: k pontos são escolhidos aleatoriamente.
 - `np.array`: matriz de $k \times d$ com k centroides.
 - `n_init`: número de vezes que o algoritmo é corrido com diferentes inicializações (`default=10`).
 - `max_iter`: número máximo de iterações (`default=300`).
 - `tol`: parâmetro que controla o critério de paragem. Valores menores resultam em mais iterações (`default=10-4`).

Clustering: *k*-Médias sklearn

O algoritmo disponibilizado pelo `sklearn` é uma versão melhorada do *k*-médias, nomeadamente em termos inicialização dos centroides.

- Os próximos comandos, carregam e inicializam um classificador *k*-médias com $k=8$ centroides.

```
from sklearn.cluster import KMeans  
kmeans=KMeans(n_clusters=8, init='k-means++', n_init=5, \  
               max_iter=500, tol=0.0001, verbose=1, \  
               random_state=42)
```

- Depois de treinar (`kmeans.fit(X)`), os resultados encontram-se em quatro atributos da função:

- `cluster_centers_`: matriz de $k \times d$ com k centroides.
- `labels_`: array com etiquetas para cada ponto.
(valores de 0 a $k-1$ indicando a pertença dos pontos aos centroides).
- `inertia_`: medida de desempenho.
- `n_iter_`: número de iterações efetuadas.

Clustering: *k*-Médias sklearn

Exemplo de quantificação de imagem:

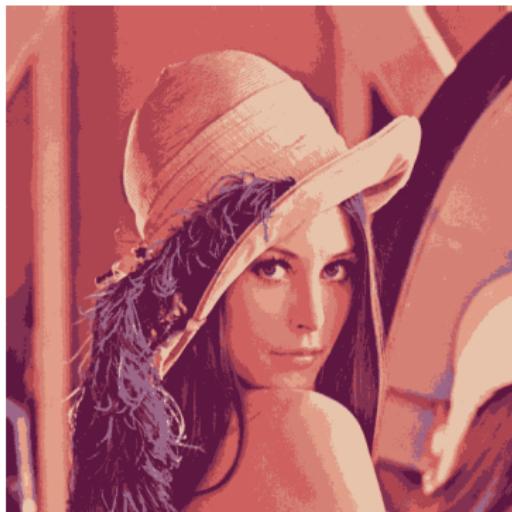
Vamos continuar o exemplo de quantificação de imagem que tínhamos iniciado, abordando os passos necessários para quantificar e visualizar a imagem resultante.

- Treinar *k*-médias e obter os pontos quantificados.

```
kmeans=KMeans(n_clusters=8).fit(X.T)
# clusters
C=kmeans.cluster_centers_
y=kmeans.labels_ # labels
# obter pontos quantificados
Xq=C[y,:]
```

- Preparar imagem e visualizar.

```
r=np.reshape(Xq[:,0],(512,512))
g=np.reshape(Xq[:,1],(512,512))
b=np.reshape(Xq[:,2],(512,512))
Iq=np.stack((r,g,b),axis=2)
Iq=np.floor(Iq).astype('uint8')
```



Clustering: Dendrogramas

Dendrogramas são árvores de relação hierárquica de proximidade entre pontos. A estratégia é juntar os dois pontos mais próximos, e repetir o processo entre pontos ou conjuntos de pontos já juntados até não haver mais pontos para juntar.

Pseudo-Código

- Conjunto de dados $\mathcal{X} = \{\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[N]\}$
 $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \rightarrow$ vetor a d dimensões.
- Inicialização: considerar cada ponto como um cluster.
(N clusters totais e cada um só com 1 ponto)
 - ▶ Encontrar os dois clusters mais próximos (mais semelhantes).
 - ▶ Juntas os dois clusters num só único.
 - ▶ Calcular a distância do novo cluster aos restantes.
 - ▶ Repetir pontos 1 a 3 até só haver um cluster.

Questões por Responder:

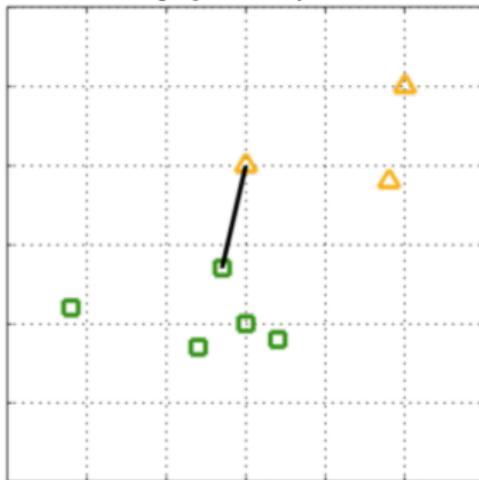
- Como calcular distâncias entre clusters?
- Como obter k centroides em vez de um só?

Clustering: Dendrogramas

Métodos de Ligação

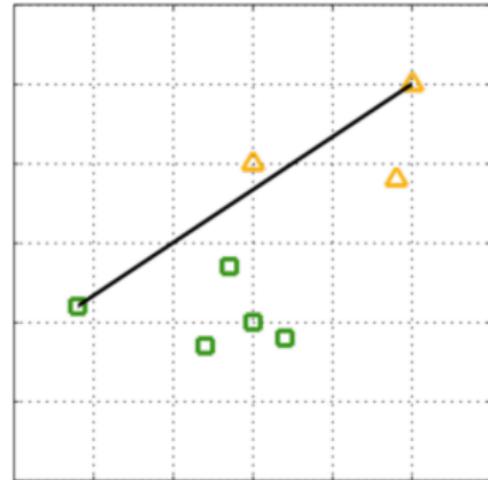
Distâncias entre clusters no clustering hierárquico são calculadas através de métodos de ligação. Existem vários destes métodos, e de seguida são dados exemplos de quatro deles.

Ligação Simples:



Menor distância entre dois pontos dos dois conjuntos.

Ligação Completa:



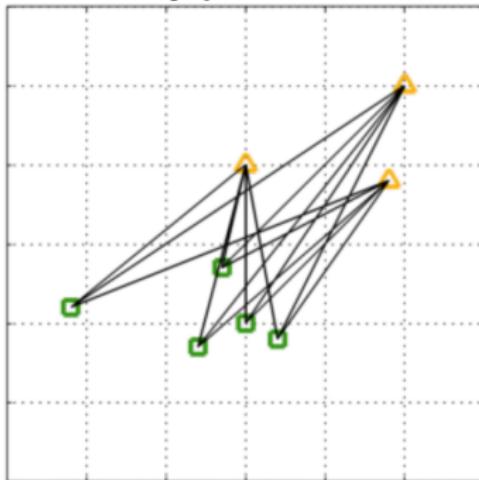
Maior distância entre dois pontos dos dois conjuntos.

Clustering: Dendrogramas

Métodos de Ligação

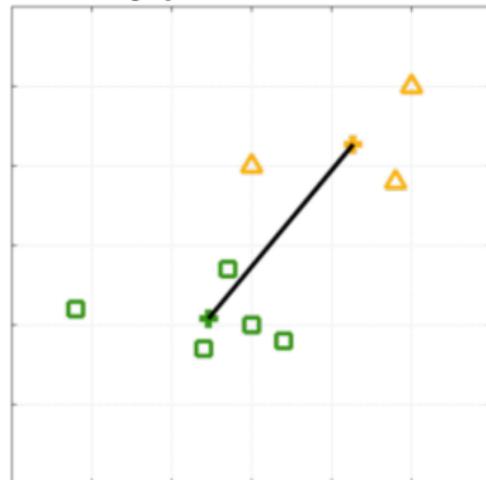
Distâncias entre clusters no clustering hierárquico são calculadas através de métodos de ligação. Existem vários destes métodos, e de seguida são dados exemplos de quatro deles.

Ligação Média:



Média das distâncias entre todos os pontos dos dois conjuntos.

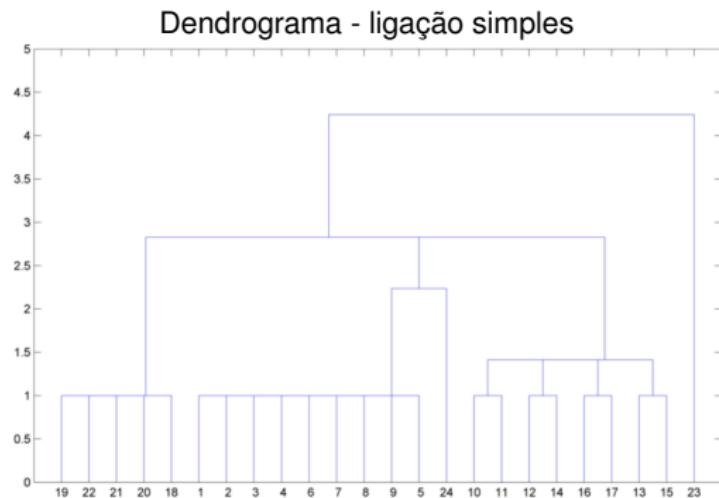
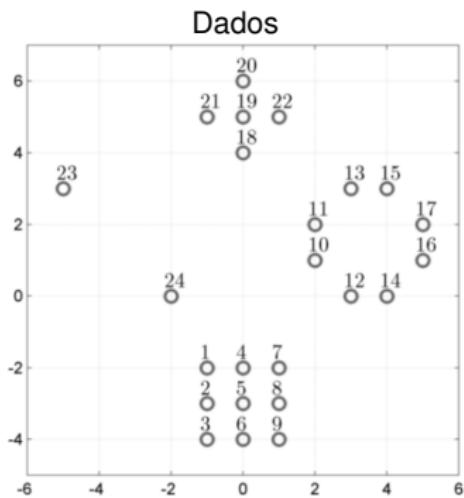
Ligação ao Centroide:



Distância entre as médias dos dois conjuntos.

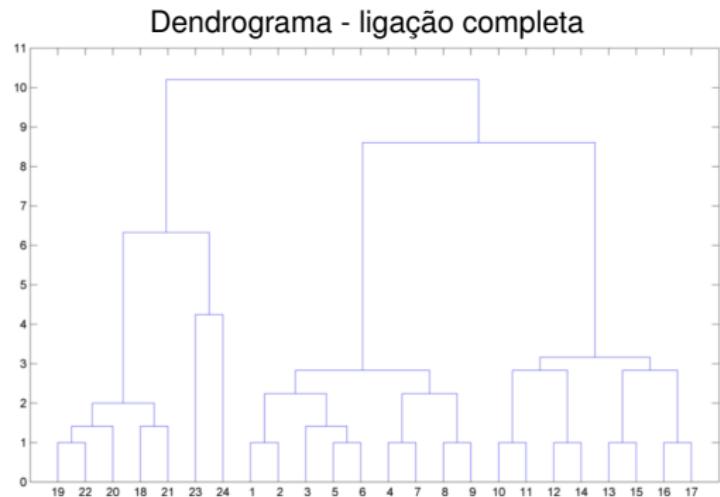
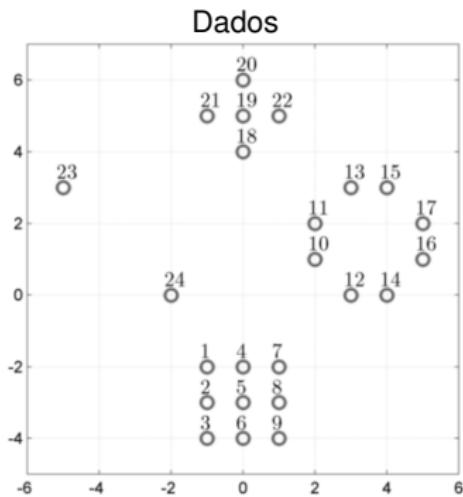
Clustering: Dendrogramas

Exemplo: dados sintéticos



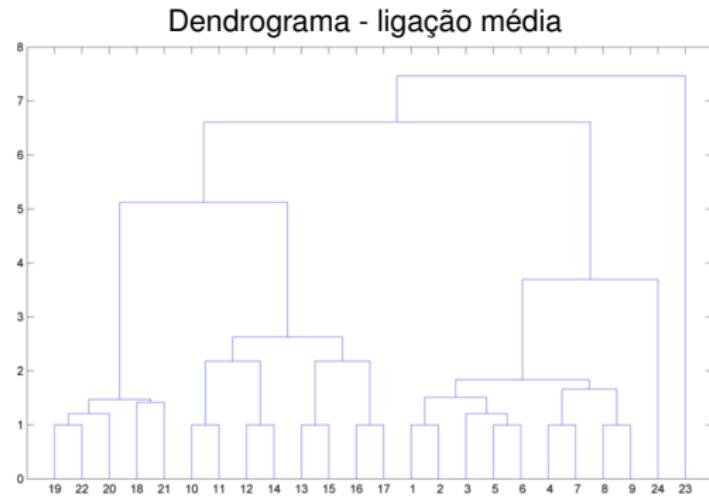
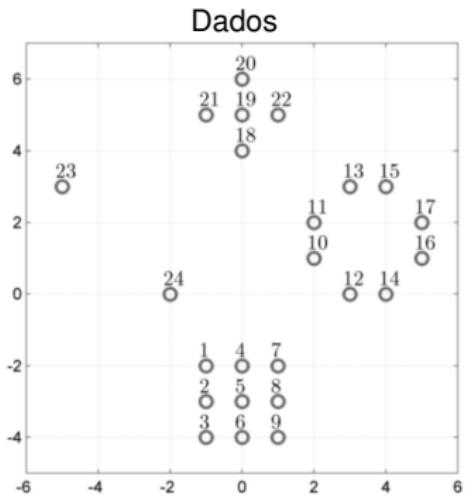
Clustering: Dendrogramas

Exemplo: dados sintéticos



Clustering: Dendrogramas

Exemplo: dados sintéticos

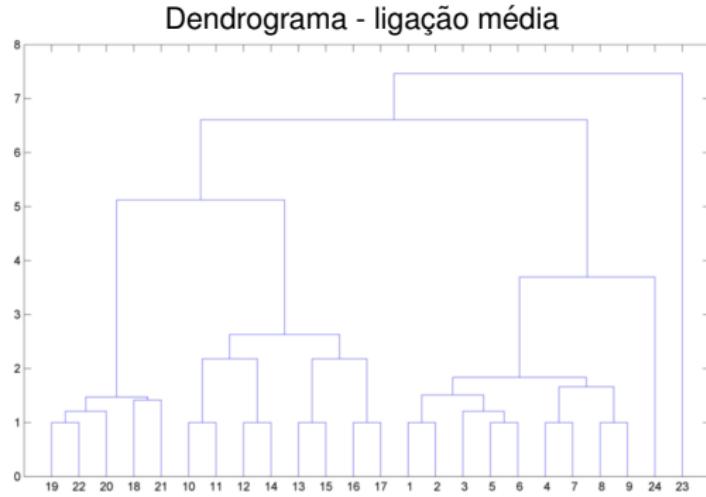
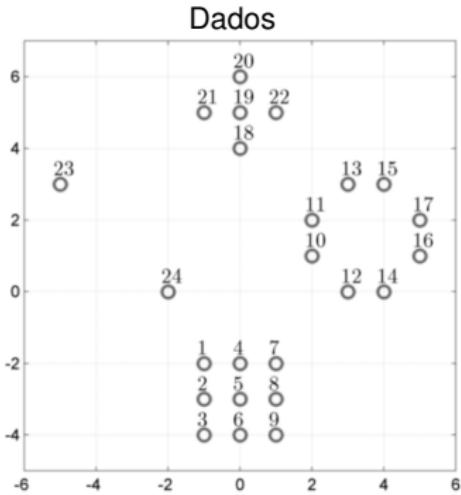


Clustering: Dendrogramas

Número de clusters

Tendo disponível o dendrograma dos pontos, pode-se visualmente aferir onde se deve “cortar” a árvore para obter os “melhores” clusters. Caso os dados não o permitam fazer, pode-se mesmo assim selecionar um número pré-definido de clusters.

Continuando o exemplo anterior de dado sintéticos, um usando o dendrograma de ligação média, obtemos os seguintes resultados.

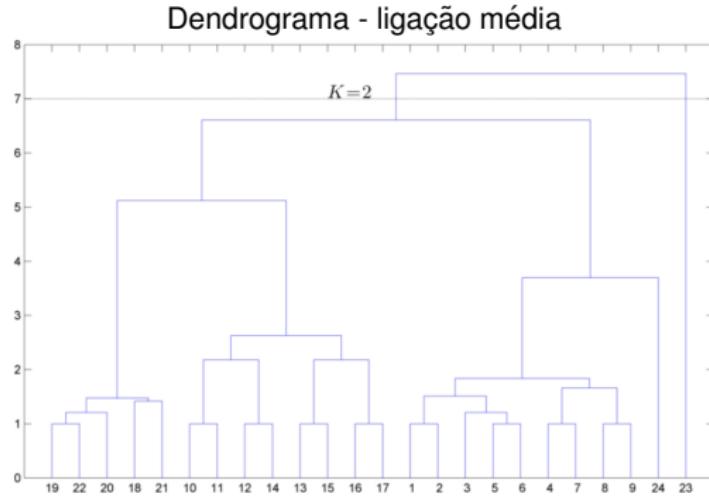
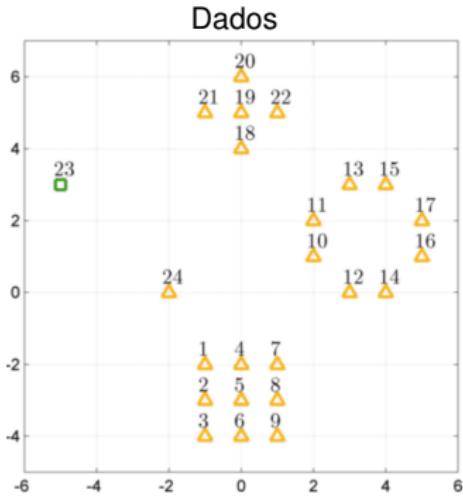


Clustering: Dendrogramas

Número de clusters

Tendo disponível o dendrograma dos pontos, pode-se visualmente aferir onde se deve “cortar” a árvore para obter os “melhores” clusters. Caso os dados não o permitam fazer, pode-se mesmo assim selecionar um número pré-definido de clusters.

Continuando o exemplo anterior de dado sintéticos, usando o dendrograma de ligação média, obtemos os seguintes resultados.

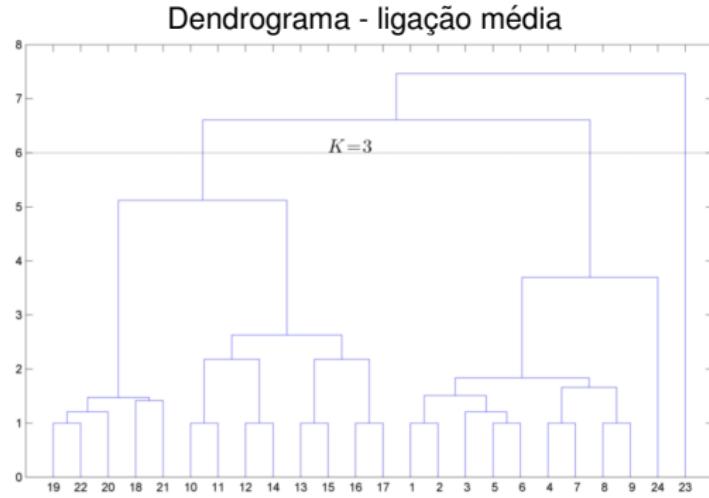
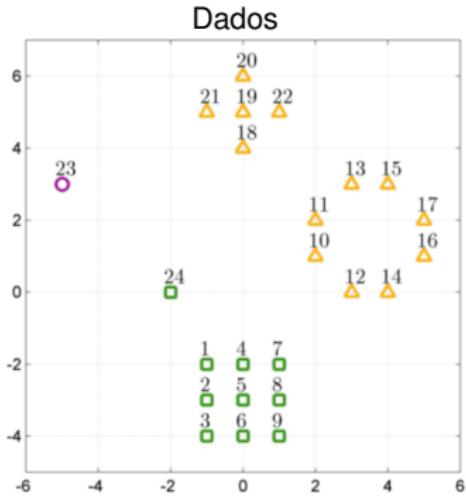


Clustering: Dendrogramas

Número de clusters

Tendo disponível o dendrograma dos pontos, pode-se visualmente aferir onde se deve “cortar” a árvore para obter os “melhores” clusters. Caso os dados não o permitam fazer, pode-se mesmo assim selecionar um número pré-definido de clusters.

Continuando o exemplo anterior de dado sintéticos, usando o dendrograma de ligação média, obtemos os seguintes resultados.

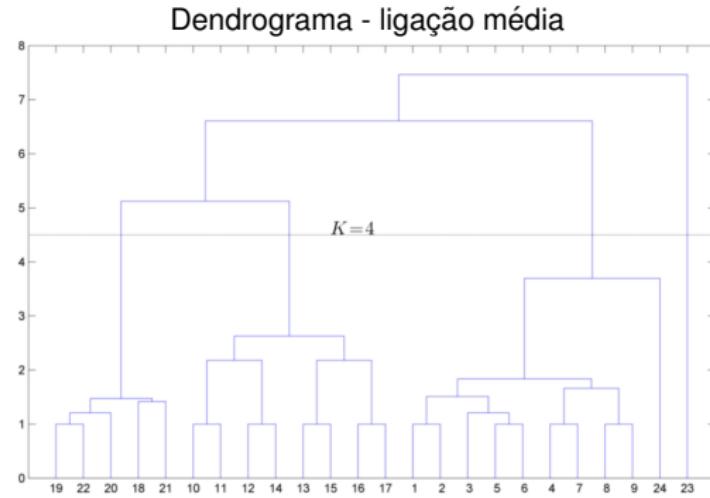
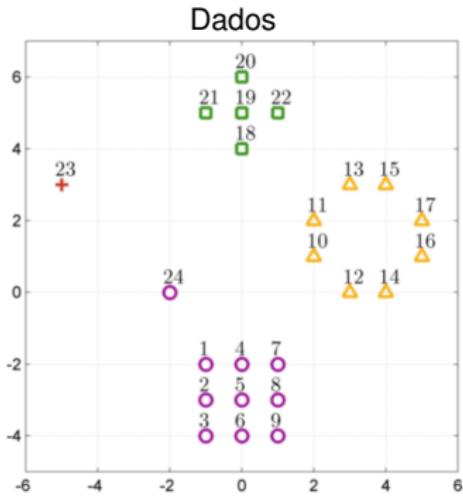


Clustering: Dendrogramas

Número de clusters

Tendo disponível o dendrograma dos pontos, pode-se visualmente aferir onde se deve “cortar” a árvore para obter os “melhores” clusters. Caso os dados não o permitam fazer, pode-se mesmo assim selecionar um número pré-definido de clusters.

Continuando o exemplo anterior de dado sintéticos, usando o dendrograma de ligação média, obtemos os seguintes resultados.

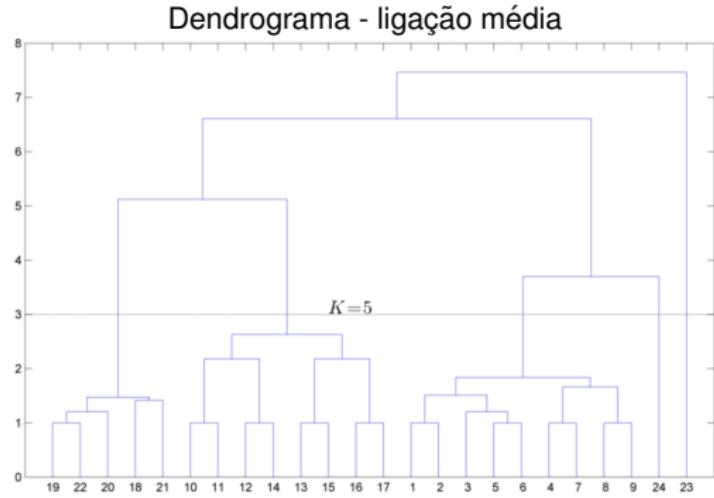
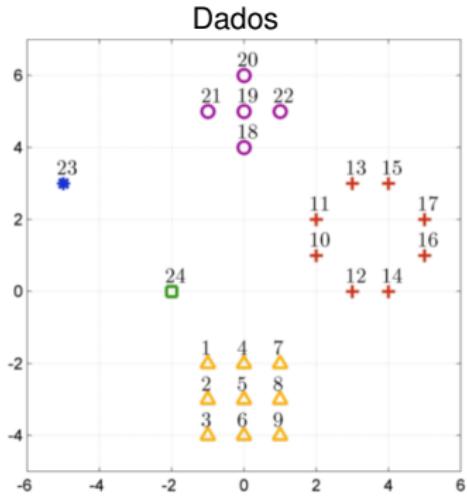


Clustering: Dendrogramas

Número de clusters

Tendo disponível o dendrograma dos pontos, pode-se visualmente aferir onde se deve “cortar” a árvore para obter os “melhores” clusters. Caso os dados não o permitam fazer, pode-se mesmo assim selecionar um número pré-definido de clusters.

Continuando o exemplo anterior de dado sintéticos, usando o dendrograma de ligação média, obtemos os seguintes resultados.



Clustering: Dendrogramas sklearn

O objeto `AgglomerativeClustering`, do sub-módulo `clustering` do `sklearn` permite criar agrupamentos através de dendrogramas.

- Os próximos comandos, carregam e inicializam um dendrograma com ligação simples e de modo a que no final haja $k=8$ centroides.

```
from sklearn.cluster import AgglomerativeClustering as aggClus
simpleLink=aggClus(n_clusters=8, affinity='l2', linkage='simple')
```

- Aqui estão alguns parâmetros a ter em conta:

- `n_clusters`: número de centroides (`default=8`).
- `affinity`: métrica de distância ('`l1`', '`l2`', '`cosine`' : `default='l2'`).
- `linkage`: método de ligação usado.
`escolhas`: '`ward`', '`single`', '`complete`', '`average`' (`default='ward'`).
- `distance_threshold`: distância de ligação acima da qual não se junta clusters (`default=None`).

Clustering: Dendrogramas sklearn

O objeto `AgglomerativeClustering`, do sub-módulo `clustering` do `sklearn` permite criar agrupamentos através de dendrogramas.

- Os próximos comandos, carregam e inicializam um dendrograma com ligação simples e de modo a que no final haja $k=8$ centroides.

```
from sklearn.cluster import AgglomerativeClustering as aggClus  
simpleLink=aggClus(n_clusters=8, affinity='l2', linkage='simple')
```

- Depois de treinar (`simpleLink.fit(X)`), os resultados encontram-se em cinco atributos da função:

- `cluster_centers_`: número de clusters encontrados.
(igual a `n_clusters` se `distance_threshold=None`)
- `labels_`: array com etiquetas para cada ponto.
(valores de 0 a $k-1$ indicando a pertença dos pontos aos centroides).
- `n_leaves_`: número de folhas da árvore.
- `n_connect_components_`: estimativa do número total de ligações.
- `children_`: as bifurcações dos nós da árvore (exceto as folhas).

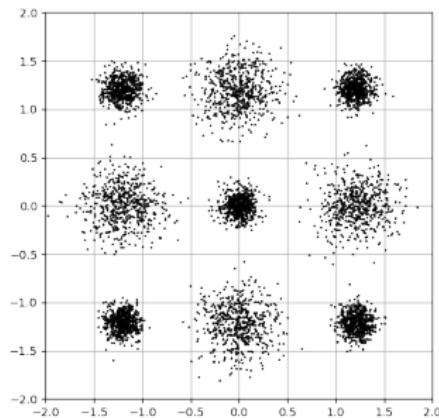
Clustering: Dendrogramas

Exemplo de dados sintéticos

- A distribuição dos dados consiste em 9 clusters gaussianos posicionados num grelha.
O número total de pontos é 5000.
Ficheiro: `mix9gausPts.p`

- Carregar dados, instanciar e treinar dendograma.

```
fName='mix9gausPts.p'  
# X matriz de 2×5000  
X=pickle.load(open(fName,'rb'))  
Link=aggClus(n_clusters=9,\n             linkage='ward').fit(X.T)  
y=Link.labels_
```



Clustering: Dendrogramas

Exemplo de dados sintéticos

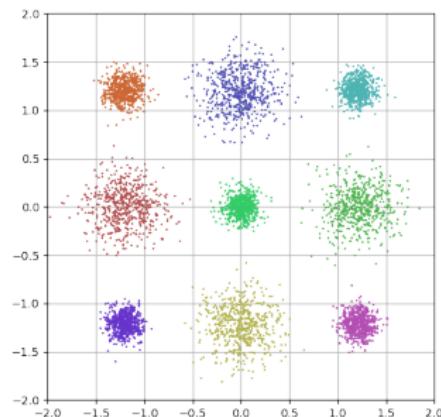
- A distribuição dos dados consiste em 9 clusters gaussianos posicionados num grelha. O número total de pontos é 5000.
Ficheiro: `mix9gausPts.p`

- Carregar dados, instanciar e treinar dendograma.

```
fName='mix9gausPts.p'  
# X matriz de 2x5000  
X=pickle.load(open(fName,'rb'))  
Link=aggClus(n_clusters=9,\n              linkage='ward').fit(X.T)  
y=Link.labels_
```

- Visualizar pontos por cluster.

```
for i in np.unique(y):  
    plt.plot(X[0,y==i],X[1,y==i],'.')
```



Clustering: Dendrogramas

Exemplo de dados sintéticos

- Para visualizar os dendrogramas pode-se usar as funções `linkage` e `dendrogram` do módulo `scipy.cluster.hierarchy`
- ou usar a função `plot_dendrogram` dos exemplos do manual do `sklearn` para visualização de dendrogramas.
- É necessário re-treinar o modelo sem um número de clusters pré-definido.

```
Link=aggClus(n_clusters=None,\n    linkage='ward', distance_threshold=0)
```

- Visualizar dendrogramas.

```
plot_dendrogram(Link,no_labels=True,\n    labels=y,color_threshold=20,\n    truncate_mode=None)
```

